Conditional Rewriting Systems for the Procedural Generation of 3D Models

Gurvan Le Guernic
Lindstedtsvägen 3, level 4, office 1433
08 790 65 40
gurvan@kth.se

October 27, 2010

Abstract

The goal of the project is to implement a 3D model viewer/navigator. The particularity of this viewer is that the 3D environment is procedurally generated. The viewer takes as input a conditional rewriting system and an initial axiom. By rewriting the initial axiom, the viewer generates the 3D environment to display.

1 Context

3D models, particularly in games, become bigger and bigger. It is common in nowadays games to see completely modeled wide cities in which the player can freely navigate. Using traditional ways of generating such cities, where designers create nearly every unique element for every new city, are not adapted for the cheap generation of such models. They are too big to be handled by a few designers in a reasonable amount of time. Another difficulty arises with online applications, such as many current games. Transferring a complete 3D model of a wide and complex area from the server to the client application is not really possible in real time. Therefore, huge 3D models are usually downloaded, or retrieved from a physical device such as a DVD, before starting to navigate the environment.

Procedural generation, and more specifically procedural modeling, is a way to handle those difficulties which has been followed, for example, by the Generative Modeling Language or Ürban PAD. Using procedural modeling, complex 3D environments are generated by evaluating a small set of rules describing how complex entities are created from smaller ones. Once this set of rules as been created, different environments can be generated by modifying the value of the initial parameters used to start the generative process. It is then easier to generate many different instances of generic environments such as forests or cities. Additionally, it is not needed anymore to transfer the whole 3D model to the client. Only the set of rules and the value of the initial parameters need to be sent to the client that will then generate the whole 3D model. Moreover, once the rules for a generic environment

have been sent, only the initial parameters need to be sent to the client to generate a new and different instance of the environment.

The goal of the project is to implement a 3D viewer that procedurally generates the 3D environment navigated from an FL-system like model. An FL-system is an L-system variant based on a conditional rewriting system à la Maude.

2 Project Description

During the project, the team should implement a software that takes as input an FL-system like model and renders the 3D environment generated from it. The GUI should allow the user to easily navigate in the model in different easily configurable modes (for example, with or without collisions, flying or not). This will involve coding a parser to retrieve the conditional rewriting rules, and a generic rewriting engine that will apply the previously parsed rewriting rules and generate the 3D model that is displayed in the GUI.

Hopefully, the project team will be able to add networking functionalities to request and retrieve FL-system models from a distant server that will also have to be coded. Additionally, the project team will attempt to improve the efficiency of the rewriting step by coding a caching mechanism.

Java is the preferred language of implementation and OpenGL the preferred graphics library. However, other languages and graphic libraries can be used. It is expected that the resulting software would be available under an open source license, but this point also can be discussed.

I am currently a postdoc researcher in the TCS group at KTH and will be able to provide assistance to the team involved in the project. I will be able to devote approximately 1 to 2 hours per week to the project team.